REMARKS

Applicants have studied the Office Action dated August 15, 2006 and have made amendments to the claims. It is submitted that the application, as amended, is in condition for allowance. Claims 1-20 are pending. Claims 1, 9, and 17 have been amended. Reconsideration and allowance of the claims in view of the above amendments and the following remarks are respectfully requested.

Claim 17 was rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Applicants have amended claim 17 in light of the specific comments of the Examiner, and submit that all pending claims are now clear and definite. Therefore, it is respectfully submitted that the rejection of claim 17 under 35 U.S.C. § 112, second paragraph, should be withdrawn.

Claims 17-20 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Applicants have amended claim 17 in light of the specific comments of the Examiner. In particular, claim 17 has been amended to recite a "computer system" that includes "a tangible storage medium for storing the plurality of configuration values in the hierarchical tree". Because the computer system includes a tangible storage medium, it cannot take the form of an entirely software embodiment. Therefore, Applicants submit that amended claim 17 is directed to statutory subject matter, and respectfully request that the rejection of claims 17-20 under 35 U.S.C. § 101 be withdrawn.

Claims 1, 2, 5-10, and 13-18 were rejected under 35 U.S.C. § 102(e) as being anticipated by Skinner et al. (U.S. Patent Application Publication No. 2003/0204517). Claims 3, 4, 11, 12, 19, and 20 were rejected under U.S.C. § 103(a) as being unpatentable over Skinner et al. in view of Hutsch et al. (U.S. Patent Application Publication No. 2001/0034771). These rejections are respectfully traversed.

The present invention is directed to an efficient and easy-to-implement method for managing configuration data. One embodiment of the present invention provides a method for managing configuration data. According to the method, configuration values are stored in a hierarchical tree having multiple nodes, a defined structure, and defined data types for the stored configuration values. Each node is associated with at least one of the configuration values, and each of the configuration values dictates how an application component associated with that configuration value behaves and/or interacts with other application components. An application component is registered with at least one of the nodes of the tree, based on a query received from the application component. The application component is notified when a configuration value associated with the at least one node of the tree is modified, based on an addition or change in at least one configuration value that matches the query.

For example, in the exemplary embodiment disclosed in the specification, configuration data, which includes configuration values, is organized into a hierarchical tree. Application components register interest in a particular node of the tree, or a particular sub-tree. Because they registered interest, these components receive notifications whenever a corresponding configuration value changes. Thus, any component that is a current holder of a registration or reference will be notified when a configuration value changes. Because the configuration values are arranged hierarchically in a tree such as an XML tree, the various application components can register for callbacks or notification upon modification of any nodes in any sub-tree.

Further, the structure of the configuration data in the XML tree can be altered, for example by adding more sub-nodes to a particular node in the tree. An interested party to the changed sub-tree will receive the new configuration data. Thus, an application component can register itself as an interested party to any change in a configuration value. Also, the use of XML to represent the tree allows a sub-tree of configuration data to be easily be expanded, with no change in how an application component handles those configuration values. Thus, application components (such as a Graphical User Interface, an object, an Application Programming Interface, a plug-in, an application, or a user) has the ability to handle changing configuration data, and can handle configuration data in a hierarchical structure such as by using the extensibility of XML.

The Skinner reference is directed toward object oriented programming and performing active update notifications within a multi-tiered application. The active update notification process of Skinner requires application components to register "interest objects". "Interest objects" indicate an interest of an application component with respect to a data object or set of data objects. The interest object is registered with an update management component of the application. The interest object specifies the interested application component, as well as the identity of one or more data objects or an attribute value or range of values to associate with data objects. When modifications are made to data objects corresponding to the registered interest objects, the interested application component or components receive an update notification from the update management component.

The interests of an application component can be further refined by registering other interest objects as sub-interests of existing interest objects. This results in an interest hierarchy within an interest registry. Skinner also teaches that interested components, also referred to as "observers," can be registered for each interest object in the hierarchy to receive notification of updates. One type of interest object collects data objects that are part of a changed data set. This allows an interested component to access those collected data objects after receipt of an update notification.

Claim 1, on the other hand, recites:

> storing a plurality of configuration values in a hierarchical tree having a plurality of nodes, a defined structure, and defined data types for the stored configuration values, wherein each node is associated with at least one of the configuration values, and each of the configuration values dictates how an application component associated with that configuration value behaves and/or interacts with other application components;
>
> registering at least one application component with at least one of the nodes of the tree, based on at least one query received from the at least one application component; and
>
> notifying the at least one application component when a configuration value associated with the at least one node is modified, based on an addition or change in at least one configuration value that matches the at least one query.

The Examiner has taken the position that “interest objects are considered to represent the configuration value”. This position of the Examiner is respectfully traversed.

Skinner explicitly teaches that an interest object “specifies the interested application component, as well as the identity of one or more data objects or an attribute value or range of values to associate with data objects”. See Skinner at Abstract. This is very different from a "configuration value" in the context of the present invention. A configuration value “dictates how an application component will behave and interact with other application components. Configuration data includes a plurality of configuration values, such as data associated with any of: user login information, an email application, an instant messaging application, a word processing application, a spreadsheet application, an image processing application or the like”. See Specification at page 1, lines 15-20. Thus, "configuration data", at least as used in the context of the present invention, is very different than the "interest objects" that are used in the system of Skinner.

The amended claims now clearly recites that each of the stored configuration values dictates how an application component associated with that configuration value behaves and/or interacts with other application components. Skinner does not teach or suggest such "configuration values", especially with respect to interest objects.

Furthermore, Skinner only teaches creating a hierarchy of interested objects and does not teach or suggest that the data is to be monitored for changes. See Skinner at Abstract; paragraphs 0071-0084; FIGs. 5A and 5B. In contrast, embodiments of the present invention store multiple configuration values in a hierarchical tree having nodes, a defined structure, and defined data types for the stored configuration values. The data is monitored for updates, and an application component that wants to be notified when a configuration value has changed is registered with a node in the hierarchical tree. Nowhere does Skinner teach this recited feature of the present invention.

Applicants believe that the differences between Skinner and the present invention are clear in amended claims 1, 9, and 17, which set forth various embodiments of the present invention. Therefore, claims 1, 9, and 17 distinguish over the Skinner reference, and the rejection of these claims under 35 U.S.C. § 102(e) should be withdrawn.

As discussed above, amended claims 1, 9, and 17 distinguish over the Skinner reference. Furthermore, the claimed features of the present invention are not realized even if the teachings of Hutsch are incorporated into Skinner. Hutsch does not teach or suggest the claimed features of the present invention that are absent from Skinner. Thus, claims 1, 9, and 17 distinguish over the Skinner and Hutsch references, and thus, claims 2-8, claims 10-16, and claims 18-20 (which depend from claims 1, 9, and 17, respectively) also distinguish over Skinner and Hutsch. Therefore, it is respectfully submitted that the rejections of claims 1-20 under 35 U.S.C. § 102(e) and 35 U.S.C. § 103(a) should be withdrawn.

If for any reason the Examiner finds the application other than in condition for allowance, the Examiner is invited to call the undersigned attorney at (561) 989-9811 should the Examiner believe a telephone interview would advance the prosecution of the application.

Respectfully submitted,

Date: November 15, 2006

By: ______________
~~Stephen Bongini~~ Jon Gibbons
~~Registration No. 40,917~~ 37,333
Attorney for Applicants

FLEIT, KAIN, GIBBONS,
GUTMAN, BONGINI & BIANCO P.L.
One Boca Commerce Center
551 Northwest 77th Street, Suite 111
Boca Raton, Florida 33487
Telephone: (561) 989-9811
Facsimile: (561) 989-9812